

LAPORAN INDIVIDU-MERGE SORT

Nama : Sayyidina Auliya

NIM : 21091397034

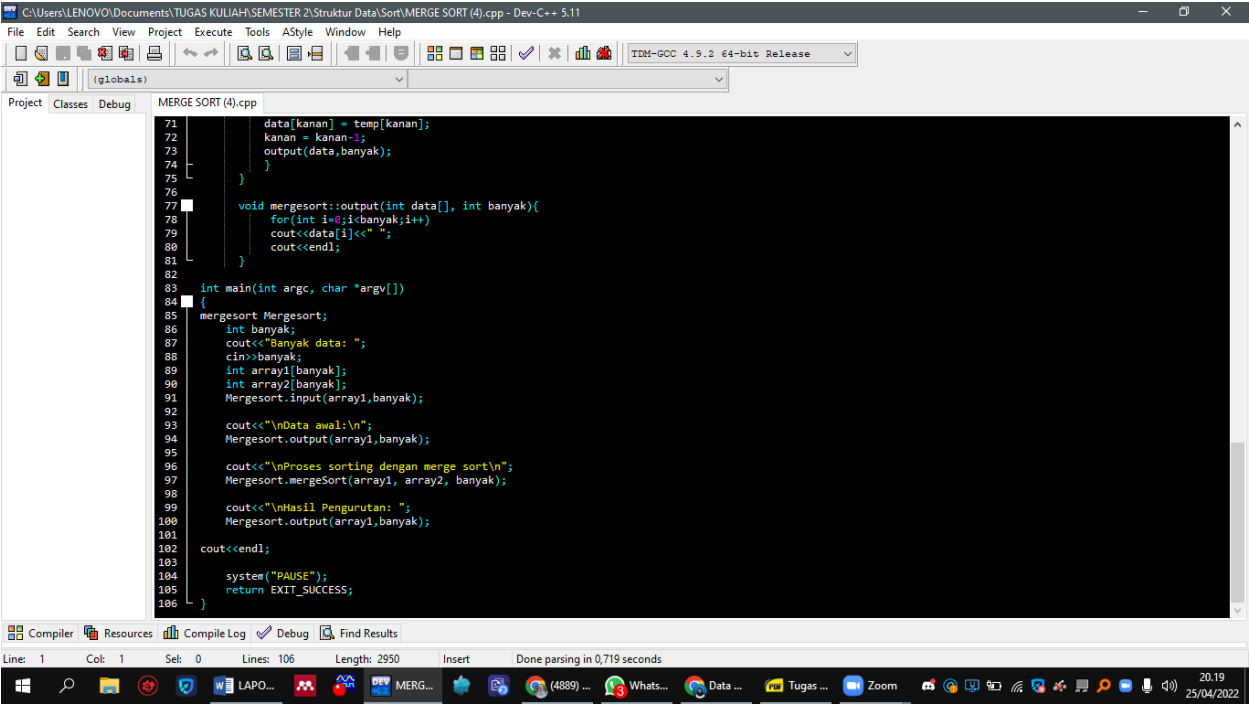
Kelas : 2021B (D4 Manajemen Informatika)

LAPORAN CODINGAN

Script merge sort :

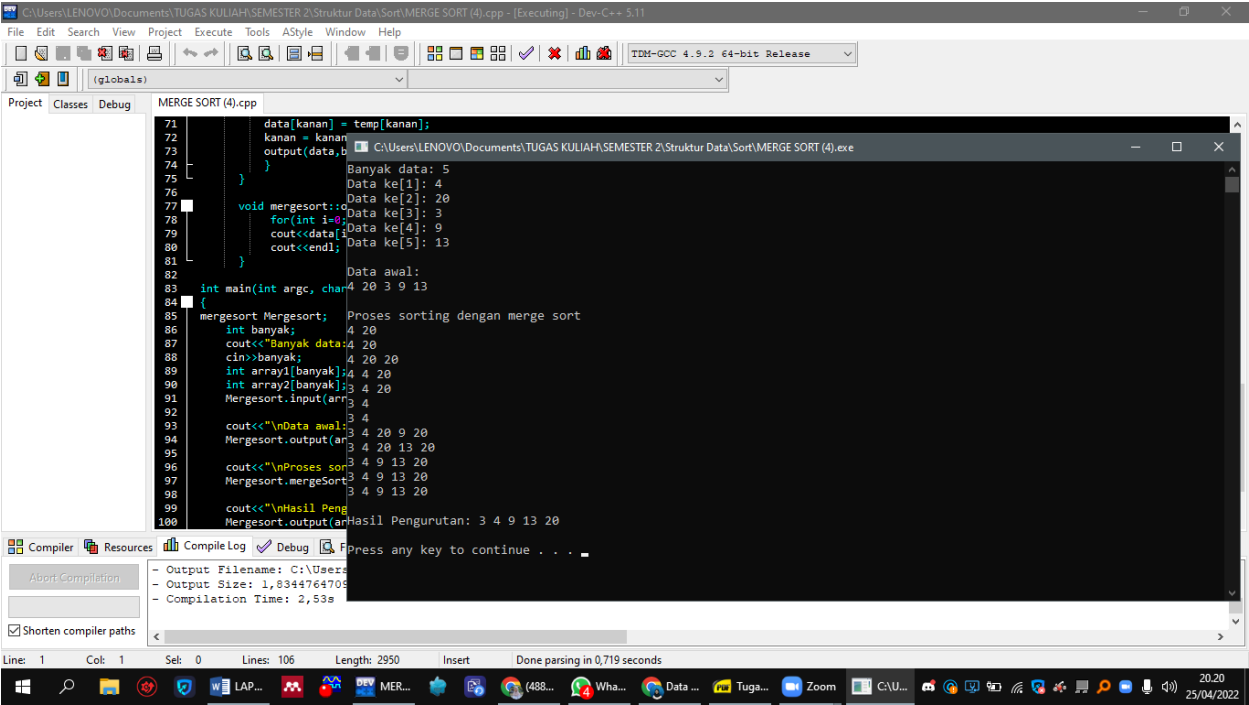
```
C:\Users\LENOVO\Documents\TUGAS KULIAH\SEMESTER 2\Struktur Data\Sort\MERGE SORT (4).cpp - Dev-C++ 5.11
File Edit Search View Project Execute Tools AStyle Window Help
(globals)
Project Classes Debug MERGE SORT (4).cpp
1 |include <csdlib>
2 |include <iostream>
3
4 using namespace std;
5 class mergesort{
6 public:
7     void input(int[], int);
8     void mergeSort(int[], int[], int);
9     void m_sort(int[], int[], int, int);
10    void merge(int[], int[], int, int, int);
11    void output(int[], int);
12
13 };
14
15 void mergesort::input(int data[], int banyak){
16     for(int i=0;i<banyak;i++){
17         cout<<"Data ke"<<i+1<<" : ";
18         cin>>data[i];
19     }
20 }
21
22 void mergesort::mergeSort(int data[], int temp[], int banyak){
23     m_sort(data, temp, 0, banyak-1);
24 }
25
26 void mergesort::m_sort(int data[], int temp[], int kiri, int kanan){
27     int tengah;
28
29     if (kanan>kiri){
30         tengah=(kanan+kiri)/2;
31         m_sort(data, temp, kiri, tengah);
32         m_sort(data, temp, (tengah+1), kanan);
33
34         merge(data, temp, kiri, (tengah+1), kanan);
35     }
36 }
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100
101
102
103
104
105
106
Length: 2950
Insert Done parsing in 0,719 seconds
20.19 25/04/2022
```

```
C:\Users\LENOVO\Documents\TUGAS KULIAH\SEMESTER 2\Struktur Data\Sort\MERGE SORT (4).cpp - Dev-C++ 5.11
File Edit Search View Project Execute Tools AStyle Window Help
(globals)
Project Classes Debug MERGE SORT (4).cpp
37
38 void mergesort::merge(int data[], int temp[], int kiri, int tengah, int kanan){
39     int bagiankiri, banyak, tmp_pos;
40
41     bagiankiri = (tengah-1);
42     tmp_pos = kiri;
43     banyak = ((kanan-kiri)+1);
44
45     while(kiri <= bagiankiri && (tengah <= kanan)){
46         if(data[kiri] <= data[tengah]){
47             temp[tmp_pos] = data[kiri];
48             tmp_pos = tmp_pos+1;
49             kiri = kiri+1;
50         }
51         else{
52             temp[tmp_pos] = data[tengah];
53             tmp_pos = tmp_pos+1;
54             tengah = tengah+1;
55         }
56     }
57     //copy bagian kiri
58     while(kiri <= bagiankiri){
59         temp[tmp_pos] = data[kiri];
60         kiri = kiri+1;
61         tmp_pos = tmp_pos+1;
62     }
63     //copy bagian kanan
64     while (tengah <= kanan){
65         temp[tmp_pos] = data[tengah];
66         tengah = tengah+1;
67         tmp_pos = tmp_pos+1;
68     }
69     //copy ke array asal
70     for(int i=0;i<banyak;i++){
71         data[kanan] = temp[kanan];
72         kanan = kanan-1;
73         output(data, banyak);
74     }
75 }
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100
101
102
103
104
105
106
Length: 2950
Insert Done parsing in 0,719 seconds
20.19 25/04/2022
```



```
71     data[kanan] = temp[kanan];
72     kanan = kanan-1;
73     output(data,banyak);
74 }
75
76
77 void mergesort::output(int data[], int banyak){
78     for(int i=0;i<banyak;i++)
79         cout<<data[i]<<" ";
80     cout<<endl;
81 }
82
83 int main(int argc, char *argv[])
84 {
85     mergesort Mergesort;
86     int banyak;
87     cout<<"Banyak data: ";
88     cin>>banyak;
89     int array1[banyak];
90     int array2[banyak];
91     Mergesort.input(array1,banyak);
92
93     cout<<"\nData awal:\n";
94     Mergesort.output(array1,banyak);
95
96     cout<<"\nProses sorting dengan merge sort\n";
97     Mergesort.mergeSort(array1, array2, banyak);
98
99     cout<<"\nHasil Pengurutan: ";
100    Mergesort.output(array1,banyak);
101
102    cout<<endl;
103
104    system("PAUSE");
105    return EXIT_SUCCESS;
106 }
```

Hasil run :



```
71     data[kanan] = temp[kanan];
72     kanan = kanan-1;
73     output(data,b C:\Users\LENOVO\Documents\TUGAS KULIAH\SEMESTER 2\Struktur Data\Sort\MERGE SORT (4).exe
74 }
75     Banyak data: 5
76     Data ke[1]: 4
77 void mergesort::o Data ke[2]: 20
78     for(int i=0 Data ke[3]: 3
79         cout<<data Data ke[4]: 9
80         cout<<endl; Data ke[5]: 13
81     }
82     Data awal:
83     4 20 3 0 13
84 int main(int argc, char
85 { mergesort Mergesort; Proses sorting dengan merge sort
86     int banyak; 4 20
87     cout<<"Banyak data:4 20
88     cin>>banyak; 4 20 20
89     int array1[banyak];4 4 20
90     int array2[banyak];3 4 20
91     Mergesort.input(ar3 4
92     3 4
93     cout<<"\nData awal: 4 20 0 20
94     Mergesort.output(ar3 4 20 13 20
95     3 4 9 13 20
96     cout<<"\nProses so3 4 9 13 20
97     Mergesort.mergeSort3 4 9 13 20
98
99     cout<<"\nHasil Peng
100    Mergesort.output(arHasil Pengurutan: 3 4 9 13 20
101
102    cout<<endl;
103
104    system("PAUSE");
105    return EXIT_SUCCESS;
106 }
```

Penjelasan algoritma proses merge sort :

[Line 5-81] proses pengurutan data :

- 1.Data yang sudah diinputkan dibagi menjadi dua sama besar, atau diambil nilai tengahnya
- 2.Larik yang sudah dibagi dua, dibagi lagi menjadi dua bagian lebih kecil
- 3.Dibagi kembali sampai menjadi bagian terkecil dan tidak bisa dibagi lagi
- 4.Mengurutkan bagian pertama
- 5.Mengurutkan bagian kedua, dst
- 6.Kemudian digabungkan

[Line 88-91] input data yang akan di urutkan dengan metode merge sort

[Line 93-94] menampilkan data awal yang sudah diinputkan

[Line 96-97] menampilkan proses sorting array dengan merge sort

[Line 99-100] menampilkan hasil sorting array dengan merge sort

KELEBIHAN DAN KEKURANGAN SORTINGAN

Kelebihan :

- sangat efisien, karena setiap list dibagi lagi menjadi list yang lebih kecil, kemudian digabungkan kembali. Sehingga tidak perlu melakukan perbandingan lagi
- cocok untuk sorting akses datanya lambat
- cocok untuk sorting data yang biasanya diakses secara berurutan

Kekurangan :

- terlalu banyak menggunakan ruang pada memori
- membutuhkan lebih banyak ruang dari pada sortingan lainnya